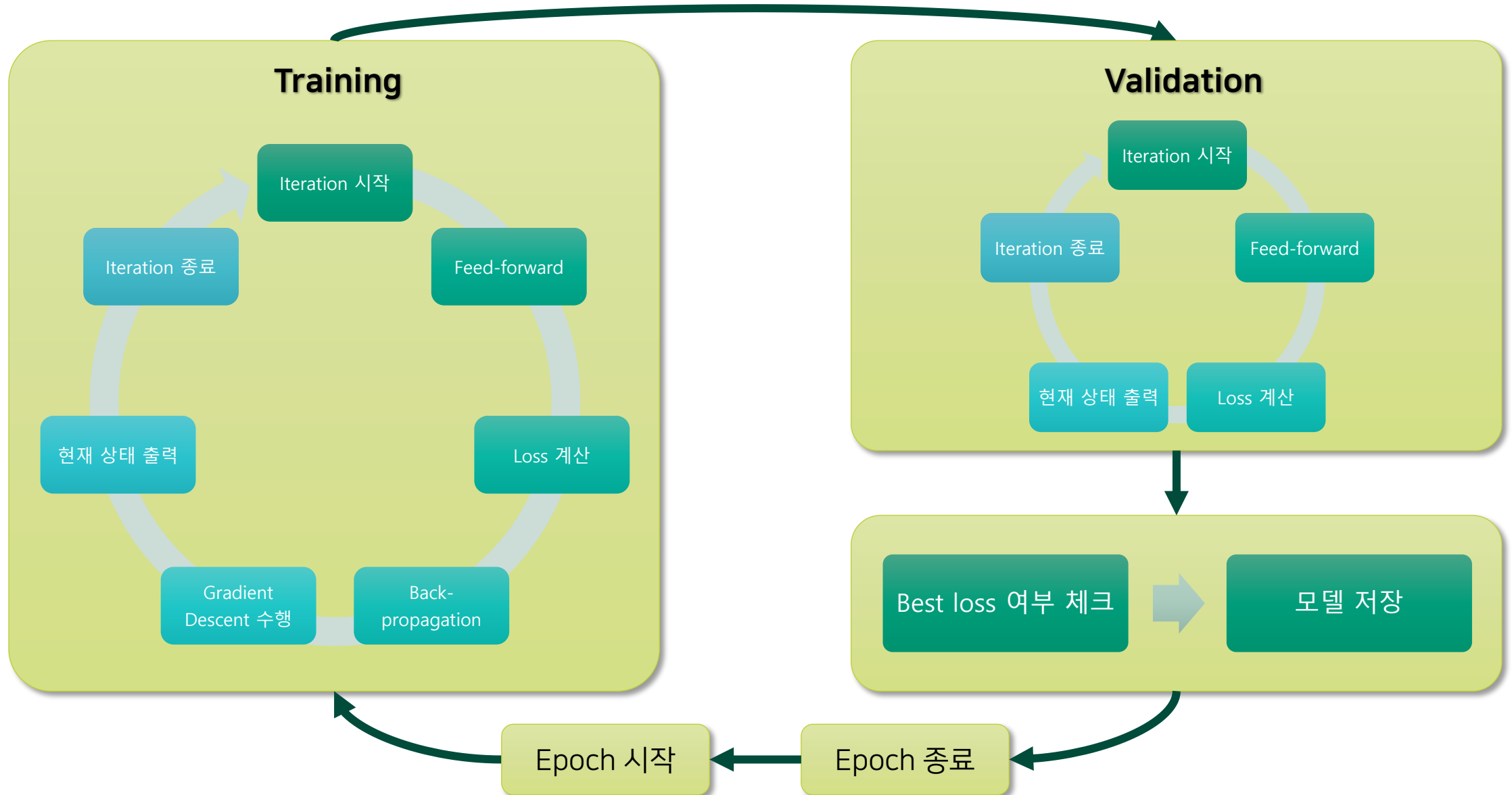


Exercise Briefing

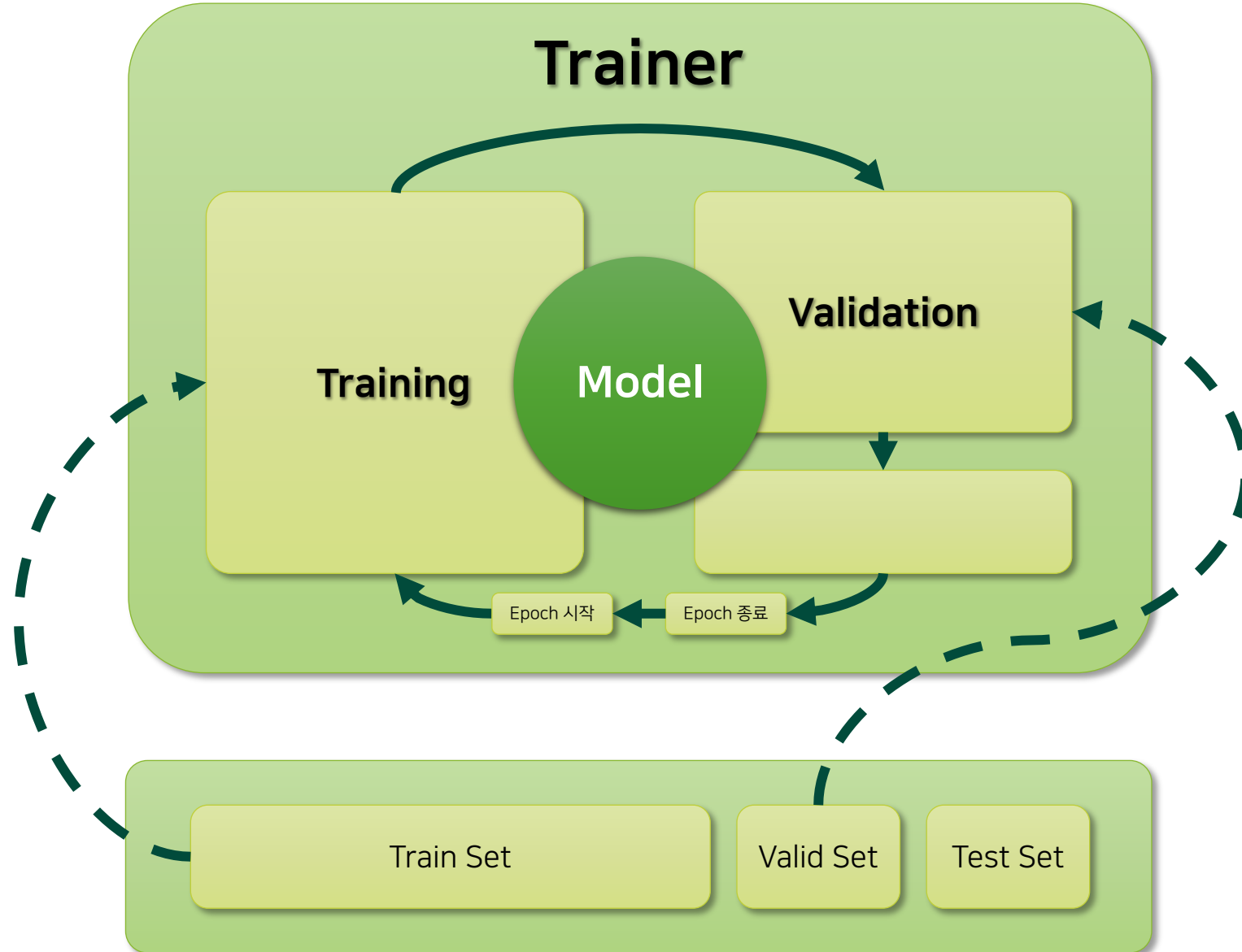
Ki Hyun Kim

nlp.with.deep.learning@gmail.com

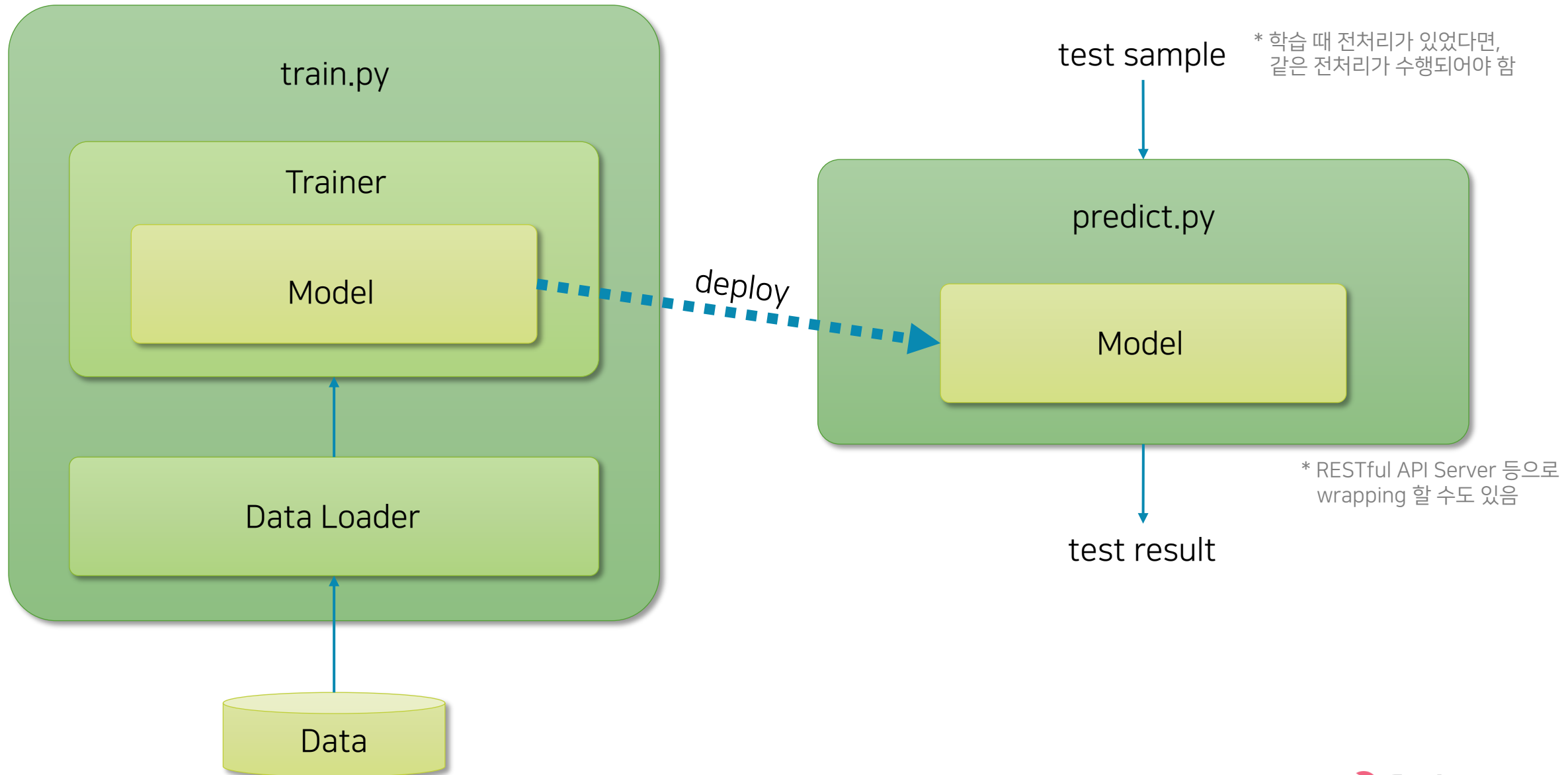
Typical Model Training Procedure



Interaction between Trainer and Others

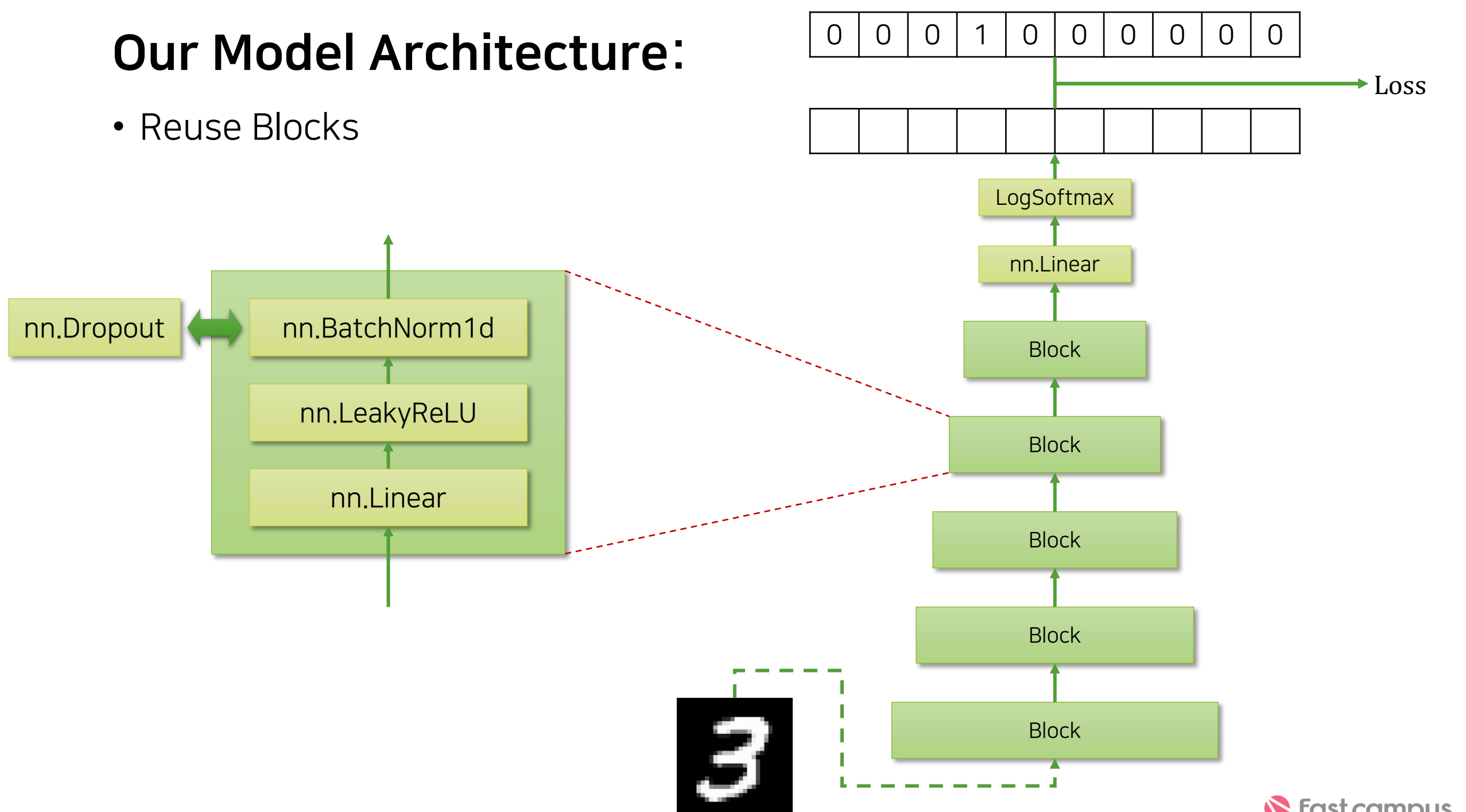


Typical Deep Learning Training / Inference Design



Our Model Architecture:

- Reuse Blocks



In Equations:

- Log-softmax + NLL Loss

$$\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$$

where $x \in [0, 1]^{N \times 28 \times 28}$ and $y \in \{0, 1\}^{N \times 10}$.

$$\log P_{\theta}(\cdot | x_i) = \hat{y}_i = f_{\theta}(x_i)$$

$$\begin{aligned}\mathcal{L}(\theta) &= -\frac{1}{N} \sum_{i=1}^N \log P(y = y_i | x = x_i; \theta) \\ &= -\frac{1}{N} \sum_{i=1}^N y_i^{\top} \cdot \hat{y}_i\end{aligned}$$

$$\hat{\theta} = \underset{\theta \in \Theta}{\operatorname{argmin}} \mathcal{L}(\theta)$$

$$\theta \leftarrow \theta - \eta \cdot \frac{\partial \mathcal{L}(\theta)}{\partial \theta}$$

Our Solution's File Structure:

- 16-practical_exercise
 - model.py
 - 모델 아키텍처 클래스를 정의
 - predict.ipynb
 - 학습이 완료된 모델 파일을 불러와, 입력을 받아 추론(inference) 수행
 - train.py
 - 사용자가 학습을 진행하기 위한 entry point
 - trainer.py
 - 모델과 데이터를 받아 실제 학습 iteration을 수행하는 클래스 정의
 - utils.py
 - 솔루션 내에서 공통적으로 활용되는 모듈들을 모아 놓은 스크립트 파일

Wrap-up: Why we are doing this

- 실무에서 초기 분석 과정을 제외한 대부분의 실험 과정은 CLI 환경에서 이루어짐
- AS-IS
 - Jupyter-notebook 활용
 - 복잡한 프로그램 구성에 불리함
- TO-BE
 - 실험을 반복해서 (+동시에) 수행할 수 있어야 함
 - 모델 아키텍처가 바뀌어도 바로 동작할 수 있어야 함
 - hyper-param를 바꿔서 다양한 실험을 돌릴 수 있어야 함
 - 코드의 일부분이 수정되어도, 다른 부분은 큰 수정이 없도록 독립적으로 동작해야 함